Information engineering for developing and testing coherent, integrated and context-dependent user interfaces

Ellemieke van Doorn¹ · Imre Horváth¹ · Zoltán Rusák¹

Received: 28 November 2016 / Accepted: 31 May 2017 / Published online: 17 June 2017
© The Author(s) 2017. This article is an open access publication

Abstract There has been an increase in the amount and complexity of traffic on Dutch inland waterways. Traffic management shifted from local traffic control to corridor traffic management. Due to these changes, operators in nautical traffic management centers experience difficulty to gain and maintain sufficient situation awareness. In our previous research, three groups of deficiencies of current situation awareness support have been identified. To overcome these deficiencies, a theoretical framework for information engineering user interface concepts was developed and used for designing three interface concepts. A coherent, an integrated and a context-dependent adaptable user interface were designed and implemented as testable prototypes. To support the process of information engineering, we combined user-centered design techniques with the formalism of set theory, directed graphs and semantic networks. From a methodological point of view, these proved to be useful both in conceptualization and implementation of the concepts. The usability of these concepts was tested in a simulator environment. The results show that the implemented prototypes represent an important improvement compared to the traditional interface solutions currently used by nautical traffic management operators. However, it was recognized that the coherent user interface implementation still suffered from a number of deficiencies. The context-dependent adaptable user interface was evaluated most positively.

Keywords User interface · Deficiencies · Information engineering · Situation awareness · Concept development · Usability testing · Coherent · Integrated · Context-dependent adaptable · Set theory · Directed graph · Semantic network

1 Introduction

The amount and complexity of traffic on Dutch inland waterways has increased and will continue to increase in the near future. To retain safe and efficient traffic flows, the focus of traffic management is shifting from local traffic control to corridor traffic management (Van Doorn et al. 2017). Consequently, traffic management operators require situation awareness (SA) of a larger area of control and a longer time span. To obtain and maintain proper SA, they rely on several information systems. Current systems, however, were not developed to support corridor traffic management. Several deficiencies of current systems for corridor traffic management have been identified (Van Doorn et al. 2015, 2017).

According to our forerunning research, three-quarter of the identified deficiencies were related to the user interfaces (UIs) and their capabilities to support human information processing. The typical reason of these deficiencies is that the system UIs display large amount of information. This makes it difficult to distinguish between relevant information about a situation and currently irrelevant information. Consequently, operators encounter difficulties in obtaining the required SA. In our research, the deficiencies related to visualization of large amounts of information were clustered in three groups; (1) the different system UIs used by the operators are not designed as a coherent whole, (2) they mainly show disjointed pieces of

✉ Ellemieke van Doorn
E.C.vanDoorn-1@tudelft.nl; Ellemieke.van.Doorn@rws.nl

¹ Faculty of Industrial Design Engineering, Delft University of Technology, Delft, The Netherlands

information about the current situation, and (3) information presentation is static, while the required SA is context-dependent.

Design of effective visualization of large amounts of information requires designers to systematically identify information needs per task (Van Doorn et al. 2017). Goal-means analysis along with abstraction hierarchy (AH) is helpful in defining relationships between information elements, which should explicitly be represented in the UI design (Ham and Yoon 2001). AH is a method proposed by Rasmussen (1986). It uses five abstraction levels to describe how the purpose of the system is implemented by which components and why certain components are needed. Abstraction modeling supports generation of high-level software design recommendations regarding system functionality based on taxonomies of usability heuristics and types of automation as well as low-level design guidelines for enhancing interface features (Kaber et al. 2006).

Though these approaches can give insights into the workflow of users and into the structure of current systems, they do not offer methods that help designers to optimize information flows within specific tasks. They facilitate the exploration of requirements, but they do not support the UI designer in modeling and specifying the UI itself (Monin and Dowell 2009). A recent attempt on extending CTA and hierarchical task analysis with failure mode and error analysis, aimed to integrate work domain, task, and social organization into a single framework to form a comprehensive human-factors analysis required for UI design (van Westrenen 2011). Although this holistic analysis framework was found to be effective in designing a maritime navigation tool, it intensively relied on the creativity of designers and did not offer a procedure to come from analysis to optimal design.

The objective of our research was to use the identified groups of deficiencies together with formal modeling to support structured information engineering in designing UI concepts. Nautical traffic management shares important characteristics with other command and control tasks, such as aviation, air traffic control, process control and traffic operations (Van Doorn et al. 2017). In all of these cases, operators work in a dynamic environment where they need to pursue multiple goals simultaneously. When developing the underpinning theories, we kept our solutions application domain independent, in order to enable their applicability in similar contexts. In this study we developed three alternative interface concepts to provide tailored solutions for the different deficiencies. This paper reports on the theoretical underpinning and the feasibility of these generic concepts. We applied the proposed underpinning theory of information engineering in developing user interfaces for nautical traffic management with the goal to describe how

we conceptualized and implemented these concepts up to a testable prototype.

In Sect. 2, we present the theories which we used for information engineering and for developing interface concepts to overcome the identified deficiencies of current systems. An application of the proposed concepts is discussed in Sect. 3 to demonstrate their feasibility. Since the implemented UI also needed to be usable by actual users in completing their tasks, we completed the usability validation of concepts. This is presented in Sect. 4. In Sect. 5 we contrast our findings with the state of the art literature, and in Sect. 6 we consolidate our results in the conclusions.

2 Interface concept development

The starting point of developing new UI concepts was the set of deficiencies that we have identified in our previous work concerning SA (Van Doorn et al. 2015, 2017). The objective was to generate proper underpinning theories for UI design concepts that can efficiently contribute to the enhancement of operators' SA. We identified three groups of related deficiencies. For each group we developed a theoretical concept aiming to overcome these deficiencies; (1) a coherent, (2) an integrated, and (3) a context-dependent adaptable UI.

2.1 Defining the coherent user interface concept

In remote control environments, software tools represent knowledge about the physical world. Van Doorn et al. (2017) showed that proper information visualization is required to allow operators to correctly access and process the information presented by the systems and to obtain the necessary SA. More specifically, the collected research data indicated that differences in the form of information visualization by the different systems used for the same tasks may lead to errors in SA. In our study, we found several deficiencies of current systems and their man-machine interface:

1. Conflicting information due to different data sources: For example, different systems showed overlapping information of which the content could conflict in case of different data sources. Data could be outdated, incomplete or incorrect.
2. Confusing information due to different ways of representing similar or the same data and information. For example, system I used a downwards pointing vertical arrow ($\downarrow$) to indicate that a vessel is traveling downstream, while system II used a left pointing horizontal arrow ($\leftarrow$) to indicate that a vessel is traveling downstream.

3. Confusing information due to similar visualization of different information. For example, system I visualized vessels carrying dangerous cargo with a red icon, while system II used a similar red icon to represent that the vessel's certificate has expired.
4. Indistinct information due to unstructured presentation of large amount of information. For instance, a long list of vessel names which required a lot of scrolling and cluttered UI which reduced SA.

To overcome the above mentioned deficiencies, we developed the concept of a coherent user interface, which we defined as follows:

A coherent user interface is a logical, consistent, orderly and harmonious interface. If there are multiple associated interfaces, then they together form a coherent whole.

A coherent UI achieves coherency in terms of the information contents and in the way of presentation of this content on the screen. We found that the information processing concerning a coherent UI can be modeled and represented by applying operations of set theory. The use of set theory enables to handle information sets to formally map pieces of information to parts of the system; determining whether or not that information entity (object) is a member of that set (information window) (Stoll 2012a). The content of every UI window can be converted to information sets per window; every interface entity related to a system can be described as the element of a set. In this way, the total information content of a system can be described as the union of all sets, $U = A \cup B \cup \dots \cup Z$, where ' $A = \{\dots\}$ ' is the subset of information entities belonging to window I. In the simplest case the system can be described by disjoint sets (sets that are not overlapping); $A \cap B \cap \dots \cap Z$ is an empty set, $\emptyset$. The same information visualization rules need to be applied to all sets to achieve a coherent interface. In cases of intersection of sets, $A \cap B \cap \dots \cap Z$ is not an empty set, additional rules are required to maintain coherency between the UI windows.

Set theory allows to mathematically codify principles of valid reasoning, and as such can be used to model how the information entities are handled by the system (Stoll 2012b). Using formal modeling based on set theory allows designers to make design decisions explicit and to maintain an overview of relationships among information entities required for design decisions. As such, it helps designers to apply their own expertise by translating design decisions to explicit rules. The rules required to achieve a coherent interface need to address each of the four aspects of coherency, i.e., logical, consistent, orderly and harmonious. These four aspects are related to coherency in terms of both information contents and the

way of information presentation. Set theory enables to formalize these aspects.

1. **Logical** A system's UI is logical, if and only if logical operations exist between the information entities of any subset of the system. Commonly used logical relations include; not ($\neg$), and ($\wedge$), or ($\vee$), if ... then ($\rightarrow$) and, if and only if ($\leftrightarrow$). For instance, if information entity Ia implies information entity Ib, so that $Ia \rightarrow Ib$, and information entity Ib implies information entity Ic, $Ib \rightarrow Ic$, then this implies that $Ia \rightarrow Ic$. For example, if wind-force '>7,' then vertical lift bridge is 'not operational' and if vertical lift bridge is 'not operational' then waterway availability is 'obstruction' infers that if wind-force '>7,' then waterway availability is 'obstruction.' If information related to obstruction of the waterway is to be highlighted, then this means that wind-force needs to be highlighted if it has a value '>7'.
2. **Consistent** A system's UI is consistent, if an only if any information entity I_k that is element of multiple subsets (A, B, ...Z), has the same information content and representation attributes in all subsets at all time. If changes to an information entity are made in window I, representations of the same information element in other window(s) are adjusted accordingly. If a red triangle has meaning 'x' in window I, then a red triangle also needs to have meaning 'x' in window II.
3. **Orderly** A system's UI is considered to be orderly, if and only if, spatial and structural arrangements of a subset of entities across and within multiple windows follow the same principles and rules. For example, information entities are listed in order of appearance; (1) vessels are listed in the order of their estimated time of arrival in the area of control, (2) traffic measures are listed in the order of their start time, and (3) log information is listed in the order of time of entry. Furthermore, all lists which are ordered based on time of appearance are displayed in the same structure. For example see Fig. 1.
4. **Harmonious** A system's UI is harmonious, if and only if, content and properties of information entities, such as syntax, semantics, color, font type, and graphical styles, are related to or complement each other. For instance, colors in the different windows have the same saturation, but do not have the same color if the information entities have a different meaning. For example see Fig. 2.

	Time	Name			Features	
a	12:00	Alex	L. 80	W. 9	D. 2.5	H. 1.2
b	12:00	Speed limit Waal			Max 5.0 m/s	
c	12:00	Maintenance Lock I			Door renewed	

Fig. 1 Orderly structure of information

Fig. 2 Example for a harmonious color use (harmonious and consistent in structure, saturation and *color*) (color figure online)

2.2 Defining the integrated user interface concept

Current nautical traffic management systems display disjointed information elements, supporting perception of separate information entities or Level 1 SA. With such a system UI, operators experience difficulties in having and maintaining a complete overview of a large area of control (Van Doorn et al. 2017). Integration (logical combination) and ultimately semantic synthesis is required to resolve this issue. Logical combination of disjointed information entities facilitates comprehension of the current traffic situation or Level 2 SA (Endsley 1995). Thus, in order to enable operators to form a holistic picture of the traffic situation and the environment, we have developed the concept of an integrated user interface, which we defined as follows:

An integrated user interface uses information fusion and capitalizes on content interactions of multiple user interface windows. Interrelated user interface contents are integrated to support specific tasks assigned to the operators' role.

An integrated UI reduces the disorganized, fragmented and redundant pieces of information as much as it is possible. Therefore, designing an integrated UI requires specifying the logical and semantic interrelationships of information entities and the functional interaction between information windows. We found that symbolic specification of these relationships needs the formalism of both set theory and common graph theory.

Integration in an integrated UI is task driven; information entities which are required for the same tasks are clustered such that each UI window supports specific tasks and/or activities. Therefore, designing an integrated UI requires defining information needs as sets per task. In Van Doorn et al. (2017), we described task analysis steps required to generate an information needs overview. As in the case of coherent interfaces, the information contents (pieces of information) required for each task and the overlap of information (i.e., the information which is needed for multiple tasks) can be specified by the means of set theory, see Fig. 3.

The use of graph theory enables to specify the functional relationships between the resultant information sets and allows to represent the fusion of information entities in an information needs graph (ING). Information entities and

Fig. 3 Information sets (A–E) per task (α – ϵ)

information subsets can be represented by vertices and the relations between them can be represented by edges of planar graphs. Various annotations can be attached to edges and the orientation of the edges can specify the direction of relationships (Newman 2010). Edges between vertices can be one-way directed, two-way symmetric directed or two-way asymmetric directed to express the nature of logical relations. This is shown in Fig. 4.

If the formalism of graph theory is used, edges can specify the nature of information fusion. For example as shown in Fig. 5, the set of vertices adjacent to vertex 9 is $N(9) = \{16, 17, D\}$. The two edges $\{16, 9\}$ and $\{17, 9\}$ are pointing toward 9. The edge $\{9, D\}$ is pointing outwards. This means that 9 is a product of 16 and 17 and that 9 is part of subset $\{D\}$. For each information entity, $\{X\}$, having edges pointing toward its vertex it needs to be defined how 'x' is calculated. For instance consider $\{9\}$ as vessel location in an hour, $\{16\}$ as current vessel location

Fig. 4 Three types of edges used in graph theory

Fig. 5 Information needs graph, representing which information entities (1–17) are required for which task (α – ϵ)

and {17} as vessel speed over ground in m/s. The element 'x1' of set {9} can be calculated using the x1 elements of set {16} and {17} with the formula $x1\{9\} = x1\{16\} + x1\{17\} \times 3600$. In this example the relation is one-way directed: {9} is not used to calculate {16} or {17}. The relation can also be two-way symmetric directed, for example if {2} is the headway, the vertical space available to allow passage under a fixed object like a non-movable bridge, {3} is the fixed height of the lower side of the object and {12} represents the water level, then $x1\{2\} = x1\{3\} - x1\{12\}$ and $x1\{12\} = x1\{3\} - x1\{2\}$. If either {2} or {12} is known, the other variable can be calculated. Both variables are possible to measure.

Consecutive edges connecting multiple vertices are called a path, which aid identification of logical groups of sets. In the example above if {2} is changed, then {12} changes which also influences {5} and {6}. Analyzing relations between subsets in the ING helps designers to look for possibilities for information fusion and for clustering these entities in a UI window or in adjacent windows. For example, the ING in Fig. 5 shows that $B \cap C = \{3, 4, 5\}$ and $B \cap C = \{2, 6, 7\}$. Paths in the ING additionally show that B and C both require availability of {12, 13, 14}. The two-way directed edges show that changing {2} or {6} results in changing {12} and as a consequence that changing {2} results in changing {6} and vice versa. This insight can be used for underpinning decision making of expert designers. In this example the ING indicates a meaningful integration of {B} and {C} in one window displaying {2, 3, 4, 5, 6, 7}. Design expertise is required to weight this and other possible design solutions. Resulting design decisions can be represented in a content integration graph (CIG). For example, integration of {B} and {C} in one window II is displayed in the CIG in Fig. 6.

If the network contains subgroups without a path between them, then each subgroup could be presented in a different UI window. For example, in Fig. 5 $A \cap E = \emptyset$, $B \cap E = \emptyset$, $C \cap E = \emptyset$, $D \cap E = \emptyset$ and $N(E) = \{10, 11\}$. Presentation of information sets {10}

and {11} in a separate window can be considered logical. See window IV in Fig. 6.

In in-between cases, when there are some relations between subgroups but not many, we considered completeness and redundancy; is it useful to display the same information twice, or can it be removed in one window when displayed in the other? Most logically information is displayed in the window with the most relations with this entity. If displayed in both windows, then the representation needs to be kept consistent. For example, in Fig. 5 $C \cap D = \{6, 7\}$ in which {6} has multiple relations with other entities related to {C} and no other relations with entities related to {D}. {7} has relations with both entities related to {C} and related to {D}. Here, it can be logical to display {6} in window II only and {7} in both windows II and III: II displaying {2, 3, 4, 5, 6, 7} and III displaying {7, 8, 9}, see Fig. 6. The presentation of {7} in the different windows needs to be consistent. Since {6} is needed for task(s) which are also supported by window III, windows II and III should be placed next to each other.

Directed edges in the CIG visualize how user's actions in an integrated UI can result in system actions in the same window and/or in other window(s). If an information entity can be adapted in a UI window, than this can be represented by a two-way symmetric directed edge between the window and the information entity vertex. Changes in information content can result in changes of related content. See {II, 2} and {II, 6} in Fig. 6. If the content of {2} is changed in window II, this affects {12} which in turn influences {6}.

In an integrated UI, user's interaction with displayed information can influence information presentation elsewhere in the UI. For example, clicking a vessel's name in one window can trigger highlighting all pieces of information that are related to this vessel in other windows. Designing an integrated UI therefore requires to identify useful interactions between windows. Interactions can be useful in cases where windows present the same information entities and/or when windows support the same task. Although all these relations are presented in the CIG, we experienced that this graph can be too complex to support a structured evaluation of relevant relations, especially in cases of information intensive systems. The relations between windows can be presented in a more clarifying way by translating the CIG representation into a windows relation network (WRN).

To represent the relations between two windows, all windows can be presented as vertices in a WRN. Weighted edges can be used to present the amount of information entities and tasks shared by two windows, see Fig. 7. The weights equal to the number of information entity vertices and task vertices that windows share in the CIG. Two windows share a vertex if they both have a path connecting them with this vertex. Paths follow the direction of the

Fig. 7 Windows relation network representations in which numbers in *italic blue* represent the weights of shared information entity vertices and the numbers in **bold red** represent the weights of shared task vertices (color figure online)

Fig. 8 Window interaction network representations of the relations between windows in an integrated UI

arrows. For example in Fig. 6 {II, 2, 12, 5} is a path, but {5, 12, 2, II} is not. The paths {7, II}/{7, III}, {14, 7, II}/{14, 7, III}, {15, 7, II}/{15, 7, III} and {16, 7, II}/{16, 7, III} represent the 4 information needs vertices shared by the windows II and III. The paths {II, δ } and {III, δ } represent the one task shared by these two windows.

The WRN representation allows to design the windows interaction graph (WIG). Each edge in the WRN represents potential useful interactions. Therefore, each edge can be evaluated by expert designers to decide whether one or more useful interactions in relation to this edge can be identified. Design decisions about useful interactions can be specified as edges in a WIG. Edges in the WIG can be any of the types shown in Fig. 4. For instance, see the simple example shown in Fig. 8. In this example, a specified user action in window I results in system action(s) in window II. User action in window II does not result in system action(s) in window I, but does result in system action(s) in windows III and IV. The interaction between windows III and IV is two-way asymmetric; a user action in window III triggers different system action(s) in window IV than a user action in window IV triggers in window III. The shown relation between windows II and IV is two-way symmetric. This means that a user action in window II triggers the same system action(s) in window IV as that this user action in window IV triggers in window II.

2.3 Defining the context-dependent adaptable user interface concept

Van Doorn et al. (2014) discussed the task and operator dependency of required SA. It was argued that changes in context influence the required SA of the operators, as well as the required support for SA. Current system interfaces, however, typically present information independent of the context. Therefore, at the development of a third alternative of supporting operator's SA, we considered a context-

dependent adaptable user interface. We defined this concept as follows:

A context-dependent adaptable user interface captures context information, assesses the implications of context, and accordingly adapts the interface content and composition to best support the pertinent tasks in the given context.

A context-dependent adaptable UI uses context-dependent decision making to adapt UI content according to the evaluated context. This means that when designing a context-dependent adaptable UI, we needed a structured information engineering approach. The context can be captured as an ordered set of information, which includes both descriptive and prescriptive elements. The descriptive elements of context describe the total of semantic relations between information entities, which play a role in a particular situation. The prescriptive elements of the context define the conditions relevant for context-dependent decision making. When modeling the constructs of context information, we found that using a semantic network (SNW) approach allows to formally and uniformly represent both the semantic relations and the decisional constraints. A SNW representation allows to map both the interface composition entities and their attributes (descriptive information entities) and the decisional constraints and assigned values (prescriptive context information entities) in a context-dependency graph (CDG).

Mathematically, a SNW is specified as a labeled directed graph of interconnected vertices. The semantic relationships between the vertices are captured by directed and labeled edges (Sowa 2014). The entire structure of the edges carries the meaning of the interconnection of information entities, represented by the vertices. Examples of relations that can be defined between vertices of a SNW are: (1) taxonomical (c is an X), e.g., 'obstruction of a waterway' is a 'priority 1 event,' (2) manifestation (1 active in duration Y), e.g., 'prognoses information' active in duration 'priority 1 event,' and (3) conditional (if $\{\neg\}a \wedge \{\neg\}b \wedge \{\neg\}c$ then $\{\neg\}X$), e.g., if there is no 'obstruction of a lock,' and no 'obstruction of a bridge' and no 'obstruction of a waterway,' then there is no 'priority 1 event,' These types of relations can be used to describe the information content, which had to be displayed (1) under a specific context condition (why), (2) at a given time (when), (3) in a relevant UI window (where), and (4) in a particular manner (how).

The specification of the interrelationships of information entities and the functional interaction between the information windows can be used as a starting point at designing a CDG. The actual information engineering (interface design) process than advances as in the case of an integrated UI (discussed Sect. 2.2, 'Integrated user

Fig. 9 Simple context-dependency graph representing a context-dependent adaptable UI (d) and the relevant parts of the related ING (a), CIG (b) and WIG (c)

interface'). Paths in the ING, CIG and WIG graphs can be used to determine which information entities and windows are relevant for a given task considering the work context. A simple example is given in Fig. 9d, which shows the CDG of a given single task, α . Let us consider this task α as 'inform skippers about restrictions due to extreme water levels.' Water levels can either be too low or too high for vessels to be able to use the concerned waterway, depending on their depth and height. The information entities 'Water level' {1}, 'Vessel depth' {3} and 'Vessel height' {4} are descriptive elements of the context with respect to task α , as it can be identified using the relevant part of the developed ING. This is shown in Fig. 9a. The pertinent CIG indicates that these entities are visualized in windows I and II, (see Fig. 9b).

Defining the context of task α also requires to prescribe what semantic relations are implied by the attributive values of the descriptive elements. For example, if water level < 0.20 N.A.P., then vessels with a depth > 0.50 m cannot use this waterway. And if water level > 1.85 N.A.P., then vessels with a height > 1.25 m cannot use this waterway. (Here N.A.P. is the reference height Normaal Amsterdams Peil, which is commonly used in the Netherlands to quantify height measurements.)

In addition to managing constraining conditions, the CDG can also be used to specify how context information has to be communicated to the user. For example, if a water level entity is < 0.20 N.A.P., then window I becomes visible. All water level entities with water level < 0.20 N.A.P. and vessel depth information of all vessels with depth > 0.50 m are highlighted in window I. And vessel icons of all vessels with a depth > 0.50 m are highlighted in window II. If the water level is > 1.85 N.A.P., then window I becomes visible. All water level entities with water level

Table 1 Definition of edges of the CDG of Fig. 9

{1, 3, a}	if $x: x < 0.20 \in \{1\} \wedge$ if $x: x > 0.50 \in \{3\}$ then {a} = active
{1, 4, a}	if $x: x > 1.85 \in \{1\} \wedge$ if $x: x > 1.25 \in \{4\}$ then {a} = active
{a, 1}	{a} = attribute of {1}
{a, 3}	{a} = attribute of {3}
{a, 4}	{a} = attribute of {4}
{1, I}	{1} is displayed in I
{3, I}	{3} is displayed in I
{4, I}	{4} is displayed in I
{a, c}	{a} = attribute of {c}
{c, II}	{c} is displayed in II
{1, b}	if $x: x < 0.20 \in \{1\}$ then {b} = active if $x: x > 1.85 \in \{1\}$ then {b} = active
{b, I}	{b} = attribute of I

> 1.85 N.A.P. and vessel depth information of all vessels with height > 1.25 m is highlighted in window I. And vessel icons of all vessels with a height > 1.25 m are highlighted in window II. The resultant CDG is shown in Fig. 9d. To improve the readability of the graph, we only visualized the type of relation between the vertices in the graph. Table 1 provides an overview of the full specification for each edge.

3 Implementation of the concepts

Implementation of the developed concepts is required to evaluate the usefulness of the concepts in practice. As a first step, we demonstrate the feasibility of applying the proposed concepts by using them to underpin the

development of UI of nautical operational network management (N-ONM) support systems. For each underpinning theory, we developed a procedure to support the development of concrete interface concepts.

3.1 Implementation of the coherent user interface concept

We used the following set theory-based procedure to realize a coherent UI at redesigning a non-coherent UI of N-ONM systems:

1. We developed a graphical user interface style guide to achieve coherency in the UI layout. The style guide defined the design of the interface elements which do not carry content information, for example, background color use and button and menu design. The total set of requirements in our style guide ensured a logical, consistent, orderly and harmonious interface. This was achieved by specifying window layout, menu structures, navigation elements, typography, and a color palette informing about what standard colors must be used throughout the interfaces.
2. We identified all UI windows of the system and defined the set of content information for each window. Set theory prescribes to use the same names to define information entities with the same meaning and to check all information entities for semantic coherence. In this step, we noticed that there were many information entities used to describe vessel information. When analyzing all these entities, we found that different descriptions were used to describe the same information element. For example, three different information elements all represented vessel type, but in a different way. ‘Official Vessel number’ (OFS), ‘Vessel number’ and ‘Vessel type.’ For example; ‘1510,’ ‘51’ and ‘Containership’ all meant the same. We translated them all to ‘Vessel type.’ The
3. We identified intersection of the content information sets, see Fig. 10. All UI windows with a geographic component (I, II and VI) share the same electronic navigational chart vector data; $A \cap B \cap F = \{\text{water, land, waterway}\}$. The information layers in window VI are also presented in window I; $F \subseteq A = \{\text{safety regions, waterway authorities, lock locations}\}$. Some information entities presented in window II are also present in windows III¹ and IV¹; $B \cap C^1 \cap D^1 = \{\text{vessel name, vessel length, vessel width, vessel height, vessel type, number of blue cones}\}$. They, however, had different data sources. These data sources also had a different update frequency, which in the current system can result in conflicting information. The information entities presented in window

Table 2 Implementation of the developed coherent user interface for N-ONM tasks

User interface windows	Related information sets: <i>such as</i>
I. Area of Focus: Static information	A. Static information with a geographic component; <i>kilometer markers, anchorage grounds, lock locations</i>
II. Area of Focus: Dynamic vessel information	B. Dynamic vessel information with a geographic component; <i>vessel location, vessel direction, vessel name, vessel type</i>
III ^x . Information overview window (multiple tabs)	C. Information overview of all entities and their main characteristics of a certain type per tab page; <i>vessel overview window displaying a list of all vessels in the area of control, their destination, dimensions, etc.</i>
IV ^x . Information details window (window per element listed in III ^x)	D. Details of one entity; <i>all relevant details on a specific vessel/event.</i>
V. Notification bar	E. Information about current notifications; <i>event title, event type, event priority level</i>
VI. Area of control	F. Static information with a geographic component; <i>safety regions, waterway authorities, lock locations</i>

Fig. 10 Coherent user interface specification based on set theory (the circles represent the information sets)

Fig. 11 Screenshot of the implemented coherent user interface

V are also presented in window III^2 ; $E \subseteq C^2 = \{\text{notification type, notification name, notification priority level}\}$. All information presented in window III is also presented in window IV; $C \subseteq D$.

4. We defined rules to handle intersection of sets and applied them to all intersecting sets. To deal with inconsistency due to different data sources, we specified that the system keeps the value which is changed most recently and thus overrides the content of all information sets in which the changed entity is present; (1) user input overrides data stored in source x, (2) in case of conflict between two sources, the most recent entry prevails.

A screenshot of the developed UI is given in Fig. 11.

3.2 Implementation of the integrated user interface concept

We used the following set theory and common graph theory-based procedure to design an integrated UI for N-ONM tasks:

1. By applying task analysis techniques, we identified five N-ONM tasks, which are presented together with the related subtasks in Table 3 (Van Doorn et al. 2017). The table includes 24 of the total of 91 information entities, which are required for these five

Table 3 N-ONM tasks and information entities per task

Tasks	Information entities
α = Assess traffic conditions	{A} = 1. Waterway, 2. Vessel ID, 3. Vessel location, 4. Vessel type, 5. Vessel direction, 6. Vessel speed, 7. Vessel destination, 8. Anchorage ground location, 9. Anchorage ground usage, 10. Event type, 11. Event location, 12. Traffic measure type, 13. Traffic measure location, 14. Log type, 15. Log content, 16. Event start time, 23 Current time
α^1 : locate/identify shipping	
α^2 : Follow shipping	
α^3 : Observe network	
α^4 : Register information	
β = Inform stakeholders	{B} = 1. Waterway, 2. Vessel ID, 3. Vessel location, 4. Vessel type, 8. Anchorage ground location, 9. Anchorage ground usage, 10. Event type, 11. Event location, 12. Traffic measure type, 13. Traffic measure location, 16. Event start time, 17. Event end time, 18. Traffic measure start time, 19. Traffic measure end time, 20. Vessel cargo, 21. Safety regions, 22. Waterway authority, 23. Current time
β^1 : provide traffic information	
β^2 : provide information in case of restrictions	
γ = Manage incidents	{C} = 2. Vessel ID, 3. Vessel location, 4. Vessel type, 10. Event type, 11. Event location, 12. Traffic measure type, 13. Traffic measure location, 20. Vessel cargo
δ = Plan traffic measures	{D} = 1. Waterway, 10. Event type, 11., Event location, 12. Traffic measure type, 13. Traffic measure location, 16. Event start time, 17. Event end time, 18. Traffic measure start time, 19. Traffic measure end time, 21. Safety regions, 22. Waterway authority, 24. Lock location
δ^1 : Determine impact of restrictions	
δ^2 : Prepare traffic measures	
ε = Set/release traffic measures	{E} = 1. Waterway, 12. Traffic measure type, 13. Traffic measure location, 14. Log type, 15. Log content, 18. Traffic measure start time, 19. Traffic measure end time. 23. Current time
ε^1 : Set traffic measures	
ε^2 : Lift traffic measures	
ε^3 : Register information

Fig. 12 Information needs graph (ING) of the developed integrated user interface for N-ONM tasks

tasks. We applied the same names to information entities having the same meaning.

- The tasks identified in the task analysis and their corresponding information subsets were represented as vertices of the ING, see Fig. 12. We used Greek symbols for identification of tasks and uppercase letters for identification of information sets: The resulting union of sets was $U = A \cup B \cup C \cup D \cup E$. The information entities were specified as a third type of vertices, which we represented using numbers. Each information entity which was part of U was represented by its own vertex. Edges between the vertices were defined to specify which information

entity is needed for which task. The information entity {9} 'Anchorage ground usage' was calculated by comparing {8} 'Anchorage ground location' with {3} 'Vessel location.' This is represented by edges pointing from {3} and {8} toward {9}. The information entities {16} 'Event start time' and {23} 'Current time' are used to decide whether planned events are already active, which is stored as status in {10} 'Event type.'

- Paths were identified in the ING to establish logical groups of sets. This step revealed that multiple tasks require the same geographic information, which best could be visualized in one map. Planning traffic

measures, however, requires geographic information of a different type and scale. This information therefore is best visualized in a separate map. The subset of 24 information entities, used as example in this paper, was arranged in 10 logical groups, shown in Table 4. These logical groups of sets and related design decisions were included in a CIG to define which information entities are presented in which UI window, see Fig. 13. Compared with the coherent UI presented in Sect. 3.1, the integrated UI had the same types of UI windows, except that the different Area of Focus windows were integrated and the information subset ‘Current time’ was placed in its own window.

4. The same graphical user interface style guide was applied to all windows of the integrated UI as which was used for our coherent UI.
5. Paths were identified in the ING and CIG to define which windows to display as adjacent windows. Several N-ONM tasks required information from multiple UI windows. Timing of tasks was carefully considered. The most time-consuming task of a N-ONM operator is task α^3 ‘Observe Network.’ The most critical N-ONM tasks

is γ ‘Manage incidents.’ Completing this task puts high demands on N-ONM operators in terms of required response times, and this task has a low error tolerance. The windows I, II and IV supporting task α^3 , needed to be displayed as adjacent windows in the system’s UI. The windows I, II and III, which support task γ , needed to be adjacent as well. The design, shown by the screenshot of the UI in Fig. 14 allowed providing sufficient support for both tasks.

6. Paths were identified in the CIG to determine shared information entities and tasks. We display these shared vertices as weighted edges in a WRN, see Fig. 15. Windows in the CIG are related if they support the same task(s) or if they display the same information entity(s).
7. We evaluated all edges in the WRN to identify useful interactions between windows. The evaluation pointed at the fact that several windows which support the same task do not share information entities. For most of these edges, our conclusion was that there is no useful interaction between the represented windows. An exception is the interaction between the windows II¹ and V. Window II¹ shows vessel names. Window

Table 4 User interface windows and information entities per window

User interface window: supported tasks	Information entities per window
I = Area of focus map: $\alpha^1, \alpha^2, \alpha^3, \beta^1, \beta^2, \gamma$	I = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}
II ¹ = Vessel overview window: $\alpha^1, \alpha^3, \beta^1, \beta^2, \gamma$	II ¹ = {2, 4, 7}
II ² = Log overview window: β^1	II ² = {14}
II ³ = Event/traffic measures overview window: $\beta^2, \delta^1, \varepsilon^2$	II ³ = {10, 12, 13, 16, 17, 18, 19}
III ¹ = Vessel detail window: β^1, γ	III ¹ = {2, 4, 7, 20}
III ² = Log detail window: α^4, ε^3	III ² = {14, 15}
III ³ = Event/traffic measures detail window: $\beta^2, \gamma, \delta^1, \delta^2, \varepsilon^1$	III ³ = {1, 10, 11, 12, 13, 16, 17, 18, 19}
IV = Notification bar: α^3	IV = {10, 12, 14}
V = Area of Control map: $\alpha^1, \alpha^3, \beta^2, \gamma, \delta^1$	V = {1, 3, 10, 11, 12, 13, 14, 21, 22, 24}
VI = Time panel: $\beta^2, \varepsilon^1, \varepsilon^2$	VI = {23}

Fig. 13 Content integration graph (CIG) of the developed integrated user interface for N-ONM tasks

Fig. 14 Screenshot of the implemented integrated user interface

Fig. 15 Window relations network (WRN) of the developed integrated user interface for N-ONM tasks

V displays the location of a vessel by showing a marker on the map. For the shared task γ ‘Manage incidents,’ we found it useful to quickly find the location of a vessel. We supported this by highlighting the location of a vessel in V when clicking this vessel’s name in II¹. With a similar objective, we evaluated all edges of the WRN. The identified useful interactions are presented in the WIG in Fig. 16 and specified in Table 5.

3.3 Implementation of the context-dependent adaptable user interface concept

When designing a context-dependent adaptable UI for N-ONM tasks, we used the following semantic network based procedure:

Fig. 16 Windows interaction graph (WIG) of the developed integrated user interface for N-ONM tasks

1. We used the results of the steps 1–7 of the procedure discussed earlier for the development of the integrated UI as the starting point for the development of a context-dependent adaptable UI for N-ONM.
2. The information acquired through task analyses and the insights from four subject-matter experts were used to identify three relevant contextual conditions. Relevant conditions are conditions which influence required SA for N-ONM tasks. The use of set theory and SNW helped to specify these conditions and to relate them to the information elements and UI windows. The identified conditions are all related to task α^3 ‘Observe network.’ Condition 1: If there is an active and/or planned event; $\{10\} \neq \emptyset$. Condition 2: If there is a currently active obstruction on the main route; $x: x_i = c \in \{10\} \wedge 11\{x_i\} \cap \{25\} \wedge 16\{x_i\} \leq \{23\}$. In which $\{25\}$ = set of coordinates representing the main route. Condition 3: If the traffic density somewhere in the area of control is equal to or larger than a threshold; $x: x \geq 10 \in \{33\}$. In which $\{33\}$ is the set of current traffic density levels and $33\{\chi\} = 2 \times n(\text{pt} \in 32(\text{GID}\chi)) + 0.1 \times n(\text{rv} \in$

Table 5 Specified interaction per edge in the window interaction graph

Edge	Interaction between user interface windows
$\{I, II^1\}$	Click vessel in I to open II^1 . Click vessel in I to highlight this vessel in II^1 Click vessel in II^1 to highlight this vessel in I
$\{I, II^2\}$	Click log in I to open II^2 . Click log in I to highlight this log in II^2 Click log in II^2 to highlight this log in I
$\{I, II^3\}$	Click event/traffic measure in I to open II^3 Click event/traffic measure in I to highlight this event/traffic measure in II^3 Click vessel in II^3 to highlight this event/traffic measure in I
$\{I, III^1\}$	Double click vessel in I to open III^1 of this vessel
$\{I, III^2\}$	Double click log in I to open III^2 of this log
$\{I, III^3\}$	Double click event/traffic measure in I to open III^3 of this event/traffic measure
$\{I, V\}$	Click vessel in I to highlight the location of this vessel in V The area displayed in I is visualized as a red rectangle in V
$\{II^1, III^1\}$	Double click vessel in II^1 to open III^1 of this vessel
$\{II^1, V\}$	Click vessel in II^1 to highlight the location of this vessel in V
$\{II^2, III^2\}$	Double click log in II^2 to open III^2 of this log
$\{II^3, III^3\}$	Double click event/traffic measure in II^3 to open III^3 of this event/traffic measure
$\{IV, II^2\}$	Click notification about log in IV to open II^2 of this log
$\{IV, III^2\}$	Double click notification about log in IV to open III^2 of this log
$\{IV, II^3\}$	Click notification about event/traffic measure in IV to open II^2 of this event/traffic measure
$\{IV, III^3\}$	Double click notification of event/traffic measure in IV to open III^2 of this event/traffic measure

$32\{GID\chi\}) + 0,1 \times n(pv \in 32\{GID\chi\}) + 1 \times n(ot \in 32\{GID\chi\})$. The definitions of the prescriptive context information entities required to define these conditions are given in Table 6.

- For each identified contextual condition, we redesigned the UI to provide support for operators to gain and maintain SA required under these circumstances. The SNW shows which information entities and windows are relevant under which conditions. This insight helped designers to specify which information to display how, where and when. In case of Condition 1 it is relevant to present an extra Area of Focus window displaying the event location; I^2 . In case of Condition 2, it is relevant to visualize the available alternative routes and the obstructed main route in Window V. In case of Condition 3 it is relevant to display traffic density information in Window V.
- We updated the CIG of step 1 with the information entities required to define the relevant context conditions, see Table 6, and added a vertex for window I^2 , see Fig. 17. I^2 displays the same information entities as I^1 and has the same interaction with the other windows as I^1 . The updated WIG (Fig. 18) together with de specification of the interaction between I^2 and the other windows is given in Table 7.
- Interface composition entities were defined to describe the total of semantic relations between information

entities and adaptable UI elements, which played a role in a particular condition. These are specified in Table 8.

- We mapped both the context conditions, the context information entities and the interface composition entities in a CDG, see Fig. 19.
 - We defined the following paths in the CDG to specify Condition 1:
 - if $\{10\} \neq \emptyset$ then $\{\alpha\} = \text{visible}$, else $\{\alpha\} = \text{not visible}$. $\{\alpha\} = \text{attribute of } \{I_2\}$
 - if $x: x_i = a \in \{10\} \vee x: x_i = c \in \{10\}$ then $11\{x_i\} = \text{coordinates (latitude, longitude) is center of map displayed in } I_2$; if $|10|$ with $x: x_i = a \in \{10\} \vee x: x_i = c \in \{10\} > 1$ then center of map displayed in $I_2 = \{11\}$ of event with lowest $|23\{x_i\} - 16\{x_i\}|$.
 - Else if $x: x_i = b \in \{10\} \vee x: x_i = d \in \{10\}$ then $11\{x_i\} = \text{coordinates (latitude, longitude) is center of map displayed in } I_2$; if $|10|$ with $x: x_i = b \in \{10\} \vee x: x_i = d \in \{10\} > 1$ then center of map displayed in $I_2 = \{11\}$ of event with lowest $|23\{x_i\} - 16\{x_i\}|$.
 - Else if $x: x_i = e \in \{10\}$ then $11\{x_i\} = \text{coordinates (latitude, longitude) is center of map displayed in } I_2$; if $|10|$ with $x: x_i = e \in \{10\} > 1$ then center of map displayed in $I_2 = \{11\}$ of event with lowest $|23\{x_i\} - 16\{x_i\}|$.

Table 6 Overview of information entities required to specify information content per condition for N-ONM tasks

Information entities	Definitions
{10} Event type	Set of events given by (type, status). Possible types are (a) obstruction complete lock, (b) obstruction partial lock, (c) obstruction waterway, (d) hindrance waterway, and (e) information event. Possible status are (0) planned and (1) active
{11} Event location	Set of (latitude, longitude) coordinates of events
{16} Event start time	Set of start times of events in yy.mm.dd.hh.mm.ss
{23} Current time	Set of current time in yy.mm.dd.hh.mm.ss
{25} Main route	Set of (latitude, longitude) coordinates of main route
{26} Availability main route	Set of availability of main route. Possible objects are: (available) and (not available)
{27} Alternative route A	Set of (latitude, longitude) coordinates of alternative route A
{28} Availability alternative route A	Set of availability of alternative route A. Possible objects are: (available) and (not available)
{29} Alternative route B	Set of (latitude, longitude) coordinates of alternative route B
{30} Availability alternative route B	Set of availability of alternative route B. Possible objects: (available) and (not available)
{31} Area of control grid	Set of (latitude, longitude) coordinates of the area of control. Each object in set {31} represents a grid identification number (GID) together with its set of coordinates: e.g., {31} can be (1, (51.72, 5.96), (51.68, 6.03)). $\cup$ = area of control
{2} Vessel ID	Set of all International Maritime Organization (IMO) numbers of vessels in the area of control. IMO is a unique reference for ships
{3} Vessel location	Set of (latitude, longitude) coordinates of all vessels in the area of control. {3} is related to {2} such that each object of {3} is given by (IMO, (latitude, longitude))
{4} Vessel type	Set of vessel type for all vessels in the area of control. Possible types are: (pt) push towing, (rv) recreational vessel, (pv) patrol vessel, and (ot) other. {4} is related to {2} such that each object of {4} is given by (IMO, type). For example (9628192, ot)
{32} Vessels in grid	Set of which vessels are in which 5 square km of the area of control. {32} is related to {31}, {2}, {3} and {4} such that each object of {32} is given by (GID, IMO, (latitude, longitude) type)
{33} Current traffic density	Set of current traffic density levels. {33} is related to {31} such that $\{33\} = \{(GID, density)\}$. Object attribute 'density' is calculated using set {32}. All object in set {32} with the same $GID(\chi)$ are taken together to calculate 'density' for this GID. $33\{\chi\} = 2 \times n(pt \subseteq 32\{GID\chi\}) + 0.1 \times n(rv \subseteq 32\{GID\chi\}) + 0.1 \times n(pv \subseteq 32\{GID\chi\}) + 1 \times n(ot \subseteq 32\{GID\chi\})$

Fig. 17 Content integration graph of the context-dependent adaptable user interface for N-ONM tasks

In prose these paths mean that if there is a planned or active event, then the extra area of focus is visible. The coordinates of the center of the map visualized in this window are the coordinates of the event. If there are multiple

events, then the event type and event start time determine which coordinates are taken as the center of the map.

b. We defined the following paths in the CDG to specify Condition 2:

Fig. 18 Window interaction graph of a context-dependent adaptable user interface for N-ONM tasks

- if $x: x_i = c \in \{10\} \wedge 11\{x_i\} \cap \{25\} \wedge 16\{x_i\} \leq \{23\}$ then $\{\beta\} = \#6E87C8$ (gray blue), else $\{\beta\} = \#11194C$ (water). $\{\beta\}$ = attribute of $\{26\}$. $\{26\}$ is displayed in V.
- if $x: x_i = c \in \{10\} \wedge 11\{x_i\} \cap \{25\} \wedge 16\{x_i\} \leq \{23\}$ then if $x: x_i = a \in \{10\} \vee x: x_i = c \in \{10\} \wedge 11\{x_i\} \cap \{27\} \wedge 16\{x_i\} \leq \{23\}$ then $\{\gamma\} = \#11194C$ (water), else $\{\gamma\} = \#A046A0$ (violet). $\{\gamma\}$ = attribute of $\{28\}$. $\{28\}$ is displayed in V.
- if $x: x_i = c \in \{10\} \wedge 11\{x_i\} \cap \{25\} \wedge 16\{x_i\} \leq \{23\}$ then if $x: x_i = a \in \{10\} \vee x:$

$x_i = c \in \{10\} \wedge 11\{x_i\} \cap \{29\} \wedge 16\{x_i\} \leq \{23\}$
 then $\{\delta\} = \#11194C$ (water), else
 $\{\delta\} = \#A046A0$ (violet). $\{\delta\}$ = attribute of $\{30\}$.
 $\{30\}$ is displayed in V.

In prose these paths mean that if there is an obstruction on the main route, then the main route is visualized in gray blue and if at that moment there is no obstruction on an alternative route, then this alternative route is visualized in violet. Else the routes are visualized in the color blue used to visualize water.

c. We defined the following paths in the CDG to specify Condition 3:

- if $x: x \geq 10 \in \{33\}$ then $\{\varepsilon\} = \text{visible}$, else $\{\varepsilon\} = \text{not visible}$. $\{\varepsilon\}$ = attribute of $\{33\}$.
- if $33\{x_i\} \leq 5$, then $\{\zeta\} = \#11194C$ (water).
- if $33\{x_i\} > 5 \wedge 33\{x_i\} \geq 10$, then $\{\zeta\} = \#8ADD50$ (green).
- if $33\{x_i\} > 10 \wedge 33\{x_i\} \geq 15$, then $\{\zeta\} = \#F7DB14$ (yellow).
- if $33\{x_i\} > 15$, then $\{\zeta\} = \#E61414$ (red).
- $\{\zeta\}$ = attribute of $\{33\}$. $\{33\}$ is displayed in V.

In prose these paths mean that if the calculated traffic density is larger than 10, then traffic density is visualized in window V. The color of a part of the waterway corresponds to the traffic density on that part of the waterway.

A screenshot of the UI is given in Fig. 20.

Table 7 Window interaction graph of a context-dependent adaptable user interface for N-ONM tasks

$\{I_1, I_2\}$: click vessel in I_1 to highlight this vessel in I_2 ; click vessel in I_2 to highlight this vessel in I_1
$\{I_2, II^1\}$: Click vessel in I to open II^1 ; click vessel in I to highlight this vessel in II^1 ; click vessel in II^1 to highlight this vessel in I
$\{I_2, II^2\}$: Click log in I to open II^2 ; click log in I to highlight this log in II^2 ; click log in II^2 to highlight this log in I
$\{I_2, II^3\}$: Click event/traffic measure in I to open II^3 ; click event/traffic measure in I to highlight this event/traffic measure in II^3 ; click vessel in II^3 to highlight this event/traffic measure in I
$\{I_2, III^1\}$: Double click vessel in I to open III^1 of this vessel
$\{I_2, III^2\}$: Double click log in I to open III^2 of this log
$\{I_2, III^3\}$: Double click event/traffic measure in I to open III^3 of this event/traffic measure
$\{I_2, V\}$: Click vessel in I to highlight the location of this vessel in V; the area displayed in I is visualized as a blue rectangle in V

Table 8 Interface composition entities and their possible values

Interface composition entity	Possible values
$\{\alpha\}$ = status of I_2	Visible Not visible
$\{\beta\}$ = color in which $\{26\}$ is displayed in V	$\#11194C$ $\#6E87C8$
$\{\gamma\}$ = color in which $\{28\}$ is displayed in V	$\#11194C$ $\#A046A0$
$\{\delta\}$ = color in which $\{30\}$ is displayed in V	$\#11194C$ $\#A046A0$
$\{\varepsilon\}$ = status of $\{33\}$	Visible Not visible
$\{\zeta\}$ = color in which $\{33\}$ is displayed in V	$\#11194C$ $\#8ADD50$ $\#F7DB14$ $\#E61414$
$\{\delta\}$ = center of map I_2	11: {latitude, longitude} of event with lowest difference between $\{23\}$ and $\{16\}$

Fig. 19 Context-dependency graph of a context-dependent adaptable user interface for N-ONM tasks

Fig. 20 Screenshot of the implemented context-dependent adaptable user interface with all conditions active

4 Testing usability of the concepts

The usability of the proposed UI concepts was validated by testing the developed prototypes in a N-ONM workplace simulator. Our objective was to test whether the proposed UIs are well suited for use in typical N-ONM tasks by a variety of N-ONM operators. Operators differ in experience and abilities, and ideally the UIs provide good support for all of them. Therefore, we aimed to involve the largest possible number of N-ONM operators in usability testing. We, however, could only involve operators who in real-life

work in the same area of control as the one which was operationalized in the simulator. In total, 60 operators were considered to be suitable to complete the testing. Of them, the company was able to free 20 operators from traffic management duties for the validation experiments. These operators were randomly selected.

4.1 Description of the usability testing environment

The developed UI prototypes have been operationalized on the N-ONM simulator, as shown in Figs. 21 and 22. A real-

Fig. 21 N-ONM workplace simulator

Fig. 22 The simulator set-up with the N-ONM workplace (on the left) and the subject-matter expert desk and observant desk (on the right)

life 24-hour log data file recorded on May 27, 2015 (a representative day in terms of availability of the waterways, and vessel types and quantity) was used in the simulator to simulate the behavior of vessels. This file contained the details of all vessels on the waterway, including voyage-, casco-, cargo- and position information for each vessel. The behavior of locks was simulated based

on log files from 2014, by considering the measured average vessel processing time per lock.

The N-ONM tasks, presented by Van Doorn et al. (2017), were used to develop three 10-minute tutorial and three one-hour traffic management scenarios. The one-hour scenarios incorporated all the identified N-ONM tasks, as shown in Table 9. The scenarios were implemented in the workplace simulator. Communication with stakeholders, such as skippers, emergency services, and colleagues, was imitated by subject-matter experts using scripts. The N-ONM operators could activate and terminate traffic measures on the simulator, which influenced the behavior of the simulated vessels and locks.

4.2 Validation of the usability testing environment

To validate whether the simulator environment was well equipped for usability testing, we assessed the completeness and realism of the environment with 12 subject-matter experts. This group of experts represented all identified stakeholders involved in the design and maintenance of traffic management information systems. Prior to a group discussion on this topic, the experts were asked to individually rate the completeness and realism of the information presented on the interfaces, using 7-point Likert scales. All subject-matter experts evaluated the simulator with at least a 4 on both aspects. The variation in their answers is given in the box plots in Fig. 23. In the group discussion, the experts unanimously agreed upon that the developed environment was elaborated enough to be used for testing the usability of the UI prototypes.

Table 9 N-ONM tasks (Van Doorn et al. 2017)

N-ONM task	Subtask—activities
Assess traffic conditions	Locate/identify shipping
	Follow shipping—interpret current traffic situation and assess current obstructions
	Observe network—upcoming traffic, anchorage grounds, congestion, restrictions
	Register information
Inform stakeholders	Provide traffic information
	Provide information in case of restrictions
Manage incidents	Take active actions for incident management—inform and coordinate actions patrol vessels
Plan traffic measure in case of restrictions	Determine impact—judge necessity traffic measures
	Prepare traffic measure (unplanned restrictions)
Set/release traffic measure (planned and unplanned restrictions)	Take actions to remedy limitations and to effect traffic measures
	Set traffic measures—notice to skippers, shipping report, activate alternative routes, coordinate actions patrol vessels
	Lift traffic measures
	Register information

Fig. 23 Box plot results of evaluation by experts

4.3 Usability testing approach

A within-subject design was used in this experiment. All operators followed the same procedure. First, the goal of the experiments was explained and all operators signed an informed consent form. After that, the three UI prototypes were presented and the operators completed three 10-min tutorial scenarios, one for each UI concept. The concepts were referred to as MMI 1 for the coherent UI, MMI 2 for the integrated UI and MMI 3 for the context-dependent adaptable UI. After ensuring their understanding of the three concepts, they executed each of the three one-hour scenarios. Counter balancing was used in combining the three different UIs with the three different scenarios, and to change the order in which the different interfaces were used. Finally, the three UI prototypes were evaluated through a structured interview. First, the interviewer asked the following questions to interrogate about the operator's experiences with the usability of the three different UI prototypes:

1. How would you describe the user interface concepts? Were they overall realistic? Did they provide proper support for the tasks at hand? Were they easy to learn?
2. To what extent the coherent user interface supported the N-ONM tasks?
3. To what extent the integrated user interface supported the N-ONM tasks?
4. To what extent the context-dependent adaptable user interface supported the N-ONM tasks?
5. Which user interface do you consider to provide the best support in terms of workload, speed of action, and the risk of making mistakes? Participants were allowed to answer with a specific MMI, with two MMI's preferred over one, or with 'no preference.'
6. Which user interface concept do you consider to provide the best support for N-ONM tasks? Participants were allowed to answer with a specific MMI, with two MMI's preferred over one, or with 'no preference.'

The operators' responses to these questions were purely subjective. We did not provide them with insights in their actual workload, speed of action and made mistakes.

After the interrogation, the operators were asked to rate the features which were used to prototype the interface concepts. Participants were asked to specify how useful they considered the various features on a five-point Likert scale from '1 = very useless' to '5 = very useful.' They read the description of the feature, as given below but in Dutch, and the interviewer showed what was meant by the description by pointing this out in a screenshot of the interfaces. For each feature which was not considered useful, the interviewer asked whether the feature could be made more useful by redesigning it, or whether the feature was considered irrelevant for the N-ONM tasks. The considered features were as follows:

1. All windows of the user interface were designed as a coherent whole (consistent information content, same map view, same use of color, same way of operation)
2. All geographic information integrated on the same map
3. Highlight location of object on the map by clicking on this object in the information overview window
4. Open detail window of object by clicking on this object on the map
5. Filter vessel overview window based on the estimated time of arrival of the vessels at a particular location
6. View type, status and location of events and notifications on the map
7. Context-dependently display a relevant location in a second area of focus window
8. Context-dependently display prognoses information in area of control map
9. Context-dependently display available alternative routes in area of control map if the main route is not available.

Feature 1 was the only feature which was active in the coherent UI prototype, known by the participants as MMI 1. Features 1–6 were active in the integrated UI prototype, named MMI 2. All features were active in the context-dependent adaptable UI prototype, MMI 3.

4.4 Results of interrogation about usability

In their response to the open questions, the operators reported that each of the three UI prototypes (1) were realistic ($n = 7$), (2) provided sufficient support for N-ONM tasks ($n = 9$), and (3) were easy to learn ($n = 9$). One operator found all three interfaces difficult to work with. Another operator found it difficult at the first time, but easier after he used the interfaces for 1 h each.

To the question concerning the extent of support provided by the implemented coherent UI for the N-ONM tasks, operators replied that they found it difficult to assess traffic conditions ($n = 7$) and to locate and identify shipping ($n = 4$) by using this interface. Some of them argued that the Area of Control window in this prototype had little added value ($n = 5$). The coherent UI was considered rather limited and difficult to work with by some operators ($n = 3$), and finding the required information with this interface was experienced as time-consuming ($n = 3$).

To the question asking about to what extent the implemented integrated UI supported N-ONM tasks, operators replied that they could quickly enough find required information ($n = 8$). Some of them replied that the interaction between windows worked well ($n = 4$). In contrast with the coherent UI, it provided good support for registering information ($n = 4$) and supported operators in locating and identifying vessels ($n = 4$). One operator reported that he did not use the interaction between the vessel information overview window and the maps. Operators considered the integration of all geographic information in one big map as easy to work with ($n = 7$). The integration of different types of information in one map according to a number of them made it easier to observe the network and to provide traffic information ($n = 4$). One operator reported that the map could also contain too much information if all information layers were made visible. In our default settings, this was not the case.

To the question concerning to what extent the implemented context-dependent adaptable UI supported N-ONM tasks, operators replied that the context-dependent visualization of an extra Area of Focus window (displaying the location of events) was considered the most valuable feature of this concept ($n = 5$). One operator, however, mentioned that this window covered information relevant in the other Area of Focus window. Two operators preferred displaying this second Area of Focus window in the Area of Control window instead of on top of the other Area of Focus window. Two operators thought that this second window did not have an added value. The presented prognosis information was considered difficult to interpret ($n = 4$). Information about traffic density was not

considered to be of use ($n = 5$). Instead, some operators preferred information about the duration of events ($n = 2$) or waiting times for locks ($n = 3$). Only one operator reported to have used the presented prognosis information and considered this valuable.

Concerning the effects of the UIs on workload, speed of actions and the risk of making mistakes, half of the operators experienced a higher workload and lower speed of actions with the coherent UI than with the other two UI (Fig. 24). Thirteen operators did not perceive any difference in terms of the risk of making errors at using the different prototypes. One operator suggested that a coherent UI could result in fewer errors than the other two interfaces. He did not experience this during the tests, but reasoned that this could be the case because this interface requires the users to memorize more, and type in names and locations, instead of clicking on icons. Users could be unaware that they clicked on the wrong icon, he argued. The argument of six other operators was that they could more easily make mistakes with the coherent user interface.

As shown in Fig. 24, none of the operators replied that they preferred the coherent UI prototype when they were asked which UI provided the best support for N-ONM tasks. Two operators indicated that the tested prototype of the integrated UI provided the best support for N-ONM tasks. Seven operators regarded the integrated UI prototype and the context-dependent adaptable UI prototype as equally suitable. Seven operators replied that the context-dependent adaptable UI prototype provides the best support for N-ONM tasks. Four operators did not have a clear preference.

The results of the evaluation of the features used to implement the UI prototypes are given in Fig. 25. This evaluation shows that all operators scored Feature 1, coherent UI design, with 3 or higher. All operators were neutral or positive about redesigning their current interface toward a coherent UI prototype. Feature 5, the possibility to filter the vessel overview window based on the estimated time of arrival of the vessels in a particular location, was evaluated as the least useful feature of the features 1–6 which together form the integrated UI prototype. Most operators scored the other features of the integrated UI prototype with a 3 or higher; they were considered useful by

Fig. 24 Pie charts evaluation of user interface concepts questions 5 and 6

Fig. 25 Box plot evaluation prototype features

most of the operators. The Feature 7, context-dependent displaying a relevant location in a second area of focus window, was the only extra feature of the adaptable UI prototype, which was considered useful by most of the operators. In addition, several suggestions were made about how to improve the usefulness of the other features of the context-dependent adaptable UI prototype. For Feature 8 (prognosis information), a total of eight suggestions for improvement were given, of which four were related to showing information about waiting times near locks. For Feature 9 (alternative routes), a total of 14 improvement suggestions were given, of which nine were related to differentiating routes per (types of) vessel(s). Analysis of the response to the open questions shows that, although the current implementation of the prognoses information and displaying of alternative routes was not considered very useful, approximately half of the operators considered these features valuable after improving their implementation.

Putting everything into the basket of validation, we could conclude that the three implemented UI prototypes were considered to be an improvement compared to currently applied user interfaces. Operators experienced several imperfections at testing the proposed coherent UI prototype. The difference between the integrated UI prototype and context-dependent adaptable UI prototype was found less significant than it was expected. Nevertheless, the context-dependent adaptable user interface prototype was evaluated as most preferably by more operators than the integrated UI. At the same time, it also showed more opportunity for improvement than the two other implemented UI prototypes.

5 Discussion

The motivation of our research was to overcome deficiencies of support for SA of current system interfaces. Other studies presented in UI literature which aim to

improve UI design commonly start with identification of novel technological affordance or powerful design concepts, which is usually followed by studies to explore their effects on the operator's SA in a specific application context. Examples of these approaches are ecological interface designs for nuclear process control (Burns et al. 2008), (Kim et al. 2012), integrated user interface designs for nurses in intensive care units (Koch et al. 2013), and interface designs for supporting the situation awareness of anesthesiologists (Zhang et al. 2002). In this paper, we discussed how we instead used the insights concerning the deficiencies of current systems and formal modeling methods used for information engineering as a robust basis of formulating concepts of efficient UIs.

We used a set theory-based approach to map information entities to the UI windows and to formalize information handling by the interfaces. Although different formal modeling techniques can also be used to specify UI, set theory is well-accepted as a robust basis for computational modeling of interfaces. For example, Duke et al. (1999) showed that formal specification techniques based on set theory can clarify what information can or should be presented to users and to specify the effect user actions should have. Bowen and Reeves (2008) proposed to formalize the meaning of user-centered design outcomes in a presentation model based on set theory. This approach ensures correct and robust implementation of design guidelines describing UI properties or design rules, such as rules for consistency.

The results shown in this paper provide evidence that set theory can be used more widely than just to specify UI design. Additionally, we used set theory to support designers in their analysis and concept development. In our case study, application of set theory helped to identify that the different maps used in current systems use different data sources for displaying vessel information than the information overview and information detail windows. This insight helped to design information handling rules to support information consistency. We also came across different information elements which all represented the same information. Input from subject-matter experts was needed to identify which data needed to be considered the same and which needed to be considered as separate information elements. After that, the same data and terminology could be used throughout the whole system. By making the overlaps explicit, set theory helped in preventing conflicting information and in preventing confusing information due to different ways of representing similar or the same data.

This paper showed that directed graphs can be used to support designing integrated UIs. As a matter of fact, using directed graphs to represent the relationships between various information entities/sets of UIs has some

preliminaries in the literature. For example, Lumertz et al. (2016) focus on modeling UI components (e.g., tabs, forms, lists) to keep consistency between their visual and behavioral characteristics. They showed that graph representation of UI contents helps maintain consistency between the interface windows of a system. They, however, do not discuss how formal modeling can support information content handling. Baumgartner et al. (2010) also use directed graphs to model relations between information entities. But in contrast to our approach, they focus on spatial and temporal relations between information entities. We focus on operator's tasks. While their proposed approach is useful to design information fusion, it does not provide support for designing content interaction of multiple user interface windows.

In our case study, the use of graph theory in information engineering helped to identify relations among information elements. The graphs revealed that two of the maps showed highly related data. For multiple tasks, the operators used both maps simultaneously. This insight helped designers to avoid or reduce displaying disjoint information elements. We integrated the information elements in a single map. Usability testing showed that operators experienced this design decision as useful or very useful. The use of graph theory also pointed toward locations for meaningful interactions between different UI windows. In our case, we found multiple relations between the information overview and information detail windows and the Area of Focus map. We therefore considered it logical to make these relations explicit in the UI design. Graph theory, however, did not help in designing the content of these interactions. For this we relied on the expertise of the designers. Operators were neutral or positive about the interactions implemented by the designers.

To model the information constructs for context-dependent adaptable UI, we used semantic network approach. As an extension of common graphs, a semantic network can represent not only connectivity, but also the semantic nature of the relationships between information constructs, which can be visualized as multi-layer space-graphs. Resembling our approach, Sottet et al. (2007) used graph representation as the basis of a task model, a concept model, and a model of the context of use. The edges between the information entities were used by them to denote mathematical rules such as context conditions. Likewise, the computational framework proposed by Motti and Vanderdonckt (2013) for a context-aware adaptable UI also uses graph representation to establishing mappings between context information and UI adaptation rules. Their framework focuses on adaptation of UIs to fit different platforms and devices. As such it is suitable to specify the UI adaptations, but it does not support context-dependent information handling. Although these different examples

could prove the advantages of using graphs to systematize the process and formalize the contents of UI design, they also cast light on the fact that different approaches can be followed in the process of operationalization.

In our study, semantic networks were used to both analyze relations between different information elements and to specify design decisions. In contrast to only using graph theory, analysis of semantic networks helped to identify meaningful interactions between different windows and aided explicit specification of context-dependent UI adaptation. This resulted in the design of three context-dependent UI adaptations. Only one of those adaptations was evaluated useful by most of the operators involved in usability testing. The comments of involved operators revealed that the implemented adaptations were considered to general. For example displaying the available alternative routes was not considered useful, while displaying alternative routes for a specific vessel was. More thorough analysis of the semantic networks might have helped to identify more specific needs for meaningful interactions between windows and context-dependent UI adaptations. Analysis of complex semantic networks, however, can be a complicated task, especially in cases of large amounts of information elements and relations between them.

In our work, we used Microsoft Visio 2010 to develop the graphs used to specify the UI. Although this tool helps to visualize graphs, it is not an efficient tool to analyze more complex semantic networks. In future work, more advanced semantic network visualization tools can be considered to better support analysis of the semantic network. Examples of semantic network visualization tools are found at www.cytoscape.org, www.ontopia.net and www.gephi.org.

6 Conclusion

The research presented in this paper had dual objectives. On the one hand, it developed a widely applicable theoretical framework of how to use formal modeling as a method to develop UI concepts. It was one of our intentions to show in this paper that insights concerning the deficiencies of current systems and formal modeling methods used for information engineering can be a robust basis of formulating concepts of efficient UIs to enhance operators' SA. On the other hand, it implemented the UI concepts in a traffic control simulator environment as testable prototypes and tested their usability with N-ONM operators in real-life scenarios. In simple words, the aim of validation was to show if and how much the three UI prototypes were able to prove the feasibility of the theoretical framework and the usability of our novel UI concepts.

Our main conclusions are as follow: As an important observation, usability testing of the implemented UI prototypes revealed that coherency alone is already an important improvement compared to the current interfaces of the traffic management systems. The results, however, also showed that the coherent UI concept still suffers from a number of deficiencies. Evaluation of the integrated UI and the context-dependent adaptable UI did not expose deficiencies in support for N-ONM tasks. Although the features of the context-dependent adaptable UI were not all considered to be useful, this concept was still evaluated most positive of all three by the team of operators. They pointed at the fact that visualization of context-dependent prognosis information was not sufficiently elaborated and gave multiple suggestions on how to make prognosis information more valuable.

This study showed that operators in general positively evaluated the developed concepts. Since the coherent UI addressed only one of the three identified groups of deficiencies, it was not surprising that this concept could not eliminate certain deficiencies with regard to supporting SA. Since the context-dependent adaptable UI is the only interface which addressed each of the three groups of deficiencies, evaluation of this prototype as the most promising was in line with our expectations. The fact that operators did not experience deficiencies when working with the integrated UI, however, raises the question whether using a context-dependent adaptable UI instead of an integrated UI will improve the SA of the operators to such extent that it warrants the consideration of the first one despite of the higher efforts and overheads of implementation. To provide input to this question, our future research will test the effects of the implemented UIs on the situation awareness of traffic operators in real-life situations.

Evaluation of our approach revealed that combined use of set theory, graph theory and semantic networks helped to (1) support identification of overlapping information and relations between information elements, (2) design information fusion and both static and context-dependent adaptable content interaction within and between UI windows, and (3) make design decisions explicit by offering a structured approach for specifying design rules. Our case study showed that the use of semantic network enabled the development of a meaningful context-dependent UI adaptation. For two of the three developed adaptations, our analysis, however, was not detailed enough to come up with meaningful adaptations. Future research should aim to better support the identification of meaningful adaptations. To fully profit from the advantages of semantic networks, we propose to consider the use of a semantic network visualization tool.

Acknowledgements This research has been supported by Rijkswaterstaat, the Executive Body of the Dutch Ministry of Infrastructure and the Environment.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Baumgartner N, Gottesheim W, Mitsch S, Retschitzegger W, Schwinger W (2010) BeAware!—Situation awareness, the ontology-driven way. *Data Knowl Eng* 69(11):1181–1193
- Bowen J, Reeves S (2008) Formal models for user interface design artefacts. *Innov Syst Softw Eng* 4(2):125–141
- Burns CM, Skraaning G, Jamieson GA, Lau N, Kwok J, Welch R, Andresen G (2008) Evaluation of ecological interface design for nuclear process control: situation awareness effects. *Hum Factors* 50(4):663–679
- Duke D, Fields B, Harrison MD (1999) A case study in the specification and analysis of design alternatives for a user interface. *Form Asp Comput* 11(2):107–131
- Endsley MR (1995) Toward a theory of situation awareness in dynamic systems. *Hum Fact* 37(1):32–64
- Ham DH, Yoon WC (2001) Design of information content and layout for process control based on goal-means domain analysis. *Cognit Technol Work* 3(4):205–223
- Kaber DB, Segall N, Green RS, Entzian K, Junginger S (2006) Using multiple cognitive task analysis methods for supervisory control interface design in high-throughput biological screening processes. *Cognit Technol Work* 8(4):237–252
- Kim SK, Suh SM, Jang GS, Hong SK, Park JC (2012) Empirical research on an ecological interface design for improving situation awareness of operators in an advanced control room. *Nucl Eng Des* 253:226–237
- Koch SH, Weir C, Westenskow D, Gondan M, Agutter J, Haar M, Liu D, Görges M, Staggers N (2013) Evaluation of the effect of information integration in displays for ICU nurses on situation awareness and task completion time: a prospective randomized controlled study. *Int J Med Inf* 82:665–675
- Lumertz PR, Ribeiro L, Duarte LM (2016) User interfaces metamodel based on graphs. *J Vis Lang Comput* 32:1–34
- Monin AE, Dowell J (2009) User interface specification with sequence diagrams: an application to the AIRBUS A380 Datalink system. *Cognit Technol Work* 11(4):267
- Motti VG, Vanderdonckt J (2013) A computational framework for context-aware adaptation of user interfaces. In: *Proceedings of the Seventh IEEE International Conference on Research Challenges in Information Science*, Paris, France
- Newman MEJ (2010) *Mathematics of networks*. In: Newman MEJ (ed) *Networks: an introduction*. Oxford University Press, Oxford, pp 109–167
- Rasmussen J (1986) *Information processing and human-machine interaction: an approach to cognitive engineering*. North-Holland, New York
- Sottet JS, Ganneau V, Calvary G, Coutaz J, Derneure A, Favre JM, Demumieux R (2007) Model-driven adaptation for plastic user interfaces. *Hum Comput Interact* 4662:397–410

- Sowa JF (2014) Issues in knowledge representation. In: Sowa JF (ed) *Principles of semantic networks: explorations in the representation of knowledge*. Elsevier, Burlington, pp 1–228
- Stoll RR (2012a) Sets and Relations. In: Stoll RR (ed) *Set theory and logic*. Dover Publications, ProQuest Ebook Central, pp 19–118
- Stoll RR (2012b) Logic. In: Stoll RR (ed) *Set theory and logic*. Dover Publications, ProQuest Ebook Central, pp 299–409
- Van Doorn EC, Horváth I, Rusák Z (2014) A systematic approach to addressing the influence of man-machine interaction on situation awareness. In: *Proceedings of the 10th International Symposium on Tools and Methods of Competitive Engineering*, Budapest, Hungary, vol. 1, pp 109–120
- Van Doorn EC, Horváth I, Rusák Z (2015) Combined use of cognitive task analysis and observational research data to identify deficiencies of support for situation awareness. In: *Proceedings of the Human Factors and Ergonomics Society 59th Annual Meeting*, Los Angeles, USA, 1717–1721
- Van Doorn EC, Rusák Z, Horváth I (2017) A situation awareness analysis scheme to identify deficiencies of complex man-machine interaction. *Int J Inf Technol Manag* 16(1):53–72
- van Westrenen F (2011) Cognitive work analysis and the design of user interfaces. *Cognit Technol Work* 13(1):31–42
- Zhang Y, Drews FA, Westenskow DR, Foresti S, Agutter J, Bermudez JC, Blike G, Loeb R (2002) Effects of integrated graphical displays on situation awareness in anaesthesiology. *Cognit Technol Work* 4(2):82–90